Use Cases for Open Cybersecurity Alliance

October 30, 2020

Doug Austin <douglas.a.austin@saic.com>

Objective:

Author three use cases for OCA's use that three use cases that would occur in the SOC and will benefit from the OCA fabric of projects.

What's in a good use case?

A good use case describes how users perform tasks. It features the user's point of view and describes the system behavior as the system responds to the user. A good use case shows sequencing and steps and reaches a specific goal.

Use Case 1: End User leans on their endpoint protection software as he experiences a phishing attack, malware is prevented.

Alice and Bob correspond often via their corporate e-mail. Bob sits down with a fresh cup of coffee to start his day and he sees a new message from Alice at the top of his inbox.

Of note – but very much going on in the background - as Bob turns on his computer, Bob's endpoint protection software starts up and checks for any new updates from the software manufacturer. A small set of signatures have been published overnight. Unknown to Bob, the signatures are installed in the background, ensuring that Bob's computer is protected from the latest known malware attacks. The software also checks in with Bob's corporate network asking a similar question, "Are there any new blocks we need to know about?" the software asks the corporate library of rules. No new rules are downloaded as no filters, blocks, proxy rules or firewall rules have been created in the last 16 hours.

Our story continues with Bob - He is eager to follow up on a project so he does not notice that the "from" name Says Alice, but it is not displaying at it typically would. He clicks anyways, as he is looking forward to the next step in a project.

Bob opens the e-mail to see "I found this link that I think will help us with our project, talk soon" – followed by a URL.

Of course Bob clicks the link, and this is what we call a phishing – or in the case of a specific user, a focused "spear phishing" attack.

The web site Bob accesses contains a zero-day cyber vulnerability. Bob did not catch it, his e-mail system's filter did not, and now Bob's endpoint software has to take over.

The link that Bob clicked opened up a few interesting tidbits, but it did not seem very apropos to Bob and Alice's project. He ignored the rest of the site, realizing – too late – that this sounded like one of those examples from his security awareness training. "I probably should not have clicked that," Bob thought. He looked again at the tempting e-mail and saw that while it had Alice's first name, it did not include a last name. And he hovered his mouse over the URL and realized, "…not a good site… uhoh."

Bob hasn't seen anything else strange, he only clicked a lick and realized quickly that it was off topic – he closed the screen so fast, at this point Bob's fears dwell and he does about his coffee – and his day.

In the background, however, the damage was done in a split second. Bob's computer saw Bob click a link – he initiated it, and Bob's computer's policies allow him to install software. The software installed itself and after setting a timer to "go off" in 4 hours, the bad program went dormant.

Malware was installed immediately as Bob clicked the URL – it all happened as the web page loaded.

Bob's endpoint protection software scanned the new program that was downloaded in the background – the software scans all files coming and going on Bob's computer.

When the new program starts to run, the virus software compares the digital signature, the certificate used to sign it, and the name, size and date on the program to its database of known malware attacks.

The new program does not match the database. The endpoint software flags it as a new, unsafe program.

Many people in Bob's company are business and mission focused – they are not savvy IT people. Because alerts have created a lot of extra calls and support costs, this security event – unknown software – does not pop up on Bob's computer.

Instead the endpoint software sends an alert to the Corporate logging tools.

In about 15 minutes an analyst will see an alert that a new file was downloaded, that it scheduled a job on Bob's computer, and that the software does not meet any known profiles for programs the corporate IT team has seen before.

The tier one security analyst confirms a recommendation to quarantine the new file. The security operations team and Bob's malware protection module on his endpoint have detected the result of Bob's click and the file has been quarantined on Bob's computer. The new file is added to the corporate database of bad software. All the endpoints in Bob's company receive this update in their twice-hourly poll of the corporate Malware protection system's database. Everyone in Bob's company who has their computer on knows within 45 minutes of Bob's login that the file he downloaded is bad. Any new experiences with this file will immediately block it, instead of waiting for the IT team to review and confirm the new policy.

Bob continues on with his day, none the wiser on how Malware is detected and endpoint protection software works. When he speaks with Alice that afternoon, he forgets about the odd e-mail from that morning, and he doesn't mention it to Alice.

Bob thinks this is the first time he has caught a cyber attack knows as "phishing" on his own, he noticed the e-mail was wrong after he looked at the web site and thought it was not very useful. The next morning, Bob sees the open e-mail, he hesitantly clicks on the header of the message in his in-box and hits the delete key, removing the already-read email from his inbox.

Use Case 2: Firewall… from the end user point of view

Following up on use case 1, above, the next day, Bob comes in to work and he has an unsolicited e-mail from a name he does not recognize. Now that Bob is a pro at spotting phishing attacks, he does not open the message and instead he clicks a button on his e-mail client that reports a bad e-mail to the corporate IT team.

The Security team sees a new alert in their system monitoring tools. The alert is titled “User Reported Malware.”

The Tier 1 security analyst reviews the new alert and analyses additional data. The Malware “catch” yesterday on Bob’s computer and today’s, plus some intelligence from sources and from the malware company, have identified a new command and control network for a cyber bot (attack) network.

The security analyst received twenty IP addresses and subnets to block, plus a new domain name to add to the corporate proxy and filters. All of these policies are implemented on a next-generation firewall that includes network decision making, packet forwarding, rule enforcement and some web proxy functions.

The Security team creates a new blocking policy and tests it on their lab systems, and rolls it out to production the same day.

The next time Bob clicks on an e-mail from these new threat actors, the web pages will not response because of the blocks implemented by the corporate security team.

Most of the time, Bob enjoys fast internet speeds and quick e-mail transfers with Alice on the exciting project they are wrapping up soon.

Use Case 3: Vulnerability Assessment Server

If you have followed our previous use cases, you know that Bob and Alice are working on a project together.

They have reached a milestone where it is time to install a server and the business software suited to their mission.

As the program manager for the implementation, Bob is responsible for making sure his team has the resources they need to be successful. His technical lead tells him the specs for a new server that is to be installed in the corporate data center with a group of users assigned to administrate the system while they install new software.

Bob submits a new server request to his IT department.

The server requests routes through three groups. A new virtual machine is built by the systems team. The windows server team receives the request and creates new settings in the corporate Active Directory. Once these two team's initial checklists are completed, the server's build process routes a new ticket to the Security Team.

A tier one analyst sees this recurring request in his team's work queue. New Server, Vulnerability Scan request.

The Analyst confirms the workflow step, right-clicking on the server's name. He picks a fuill-featured vulnerability scan from a context sensitive list in his SOAR tool.

The action creates a job on the vulnerability server.

25 minutes later, the deep scan – with credentials – of the server is completed. A report is returned to the tier one SOC analyst. One new patch is required.

The analyst returns the results to the Windows Systems team who sent the job over about thirty minutes before.

Because this is not a production system yet, they can patch it during the day. The Windows team routes a patch/update job and the new patch is downloaded.

The workflow returns the system to the security team for another scan since the last one was not successful.

Another scan request is issued, the vulnerability scanner compares the latest database of known patches, issues and vulnerabilities to its corporate master database. The scan result is clean.

A successful scan report is returned to the security team. They clear the system for use and the ticket's teams all weigh in and concur the system is ready for use.

About four hours after submitting the very-detailed request to the IT team, Bob receives an e-mail with the system name and an admin user name – not Bob's. The administrator is a member of his team who will perform the software installation. The scan report seems technical to Bob, he skims it but does not really understand it.

Bob's technical lead sees that a new .NET patch was installed that day and he records this – one less step his team will have to perform, and it may create a new testing scenario for the software installation specialists. He makes a note to mention this on his next team call and he send the server information over to the administrator who will begin the software install tomorrow.

Bob and his team really don't see how much automation was working in the background, but they know they are a step closer to system acceptance because they have a "green" vulnerability scan.

That night the vulnerability scan team adds the new server to their regular rotation of systems that the vulnerability scanning tool checks each night.